IN THE CLAIMS

Please amend claims 1-3, 5-8, 11, 14, 19, and 24 as indicated below.

1. (Currently Amended) An apparatus comprising:

a processor;

a graphics unit;

a memory coupled to the processor and the graphics unit, wherein the memory is configured to store processor data and graphics unit data; and

a shared cache coupled to the processor, the graphics unit, and the memory, wherein the shared cache is configured to store processor data and graphics unit data;

wherein the graphics unit is configured to:

partition images to be rendered into a plurality of subset areas;

track the number of times data corresponding to each of the subset areas is considered during the rendering of a first image; and

determine for each of the subset areas whether ~~the~~ data corresponding to a subset area is cacheable.

2. (Currently Amended) The apparatus as recited in claim 1, wherein data corresponding to a subset area is determined to be cacheable if the number of times data corresponding to the subset area was considered during the rendering of the first image exceeds a threshold.

3. (Currently Amended) The apparatus as recited in claim 1, wherein data corresponding to a subset area is determined to be cacheable if the number of times data corresponding

to the subset area was considered during the rendering of the first image was high relative to other subset areas of the image.

4. (Original) The apparatus as recited in claim 2, wherein the graphics unit includes an entry for each of said subset areas, each entry including a count indicative of a number of times data corresponding to the subset area is considered during the rendering of an image.

5. (Currently Amended) The apparatus as recited in claim 4, wherein the graphics unit is further configured to:

reset the count values for each of the subset area entries prior to rendering a new image; and

increment an entry count in response to considering data corresponding to said subset area entry.

6. (Currently Amended) The apparatus as recited in claim 1, wherein the graphics unit includes a graphics unit cache configured to store graphics data, and wherein during the rendering of a second image, the graphics unit is further configured to store data evicted from the graphics unit cache in the shared cache only if an indicator which corresponds to the data evicted from the graphics unit cache indicates the data evicted from the graphics unit cache is cacheable, said indicator being set prior to initiating the rendering of the second image.

7. (Currently Amended) The apparatus as recited in claim 4, wherein the count is indicative of whether or not the corresponding subset area data is cacheable.

8. (Currently Amended) The apparatus as recited in claim 4, wherein the entry for each of the subset areas further includes a flag indicative of whether or not the corresponding subset area data is cacheable.

9. (Original) The apparatus as recited in claim 2, wherein the threshold is programmable.

10. (Original) The apparatus as recited in claim 2, wherein the value of the threshold is responsive in part to the processor data miss rate in the shared cache.

11. (Currently Amended) A method for managing graphics data, said method comprising:

partitioning a first image to be rendered into a plurality of subset areas;

storing a count corresponding to a first entry of a plurality of entries in response to considering graphics data corresponding to a subset area which corresponds to said first entry; and

providing an indication corresponding to said first entry subsequent to rendering said first image, wherein said indication indicates whether or not data corresponding to a subset area which corresponds to said first entry is cacheable in a shared cache.

12. (Original) The method as recited in claim 11, wherein said indication indicates said first entry is cacheable if the count corresponding to the subset area exceeds a threshold.

13. (Original) The method as recited in claim 11, wherein said indication indicates the subset area which corresponds to said first entry is cacheable if the count corresponding to the subset area is high relative to counts corresponding to the remaining subset areas of said plurality of subset areas.

14. (Currently Amended) The method as recited in claim 12, further comprising:

initiating a fetch of said graphics data, in response to determining said graphics data is not resident in a graphics unit;

determining whether resident graphics data is to be evicted in order to store said fetched graphics data; ~~and~~

storing evicted data in a shared cache in response to detecting a flag corresponding to said evicted data indicates said evicted data is cacheable; and

storing the evicted data in a memory without storing the evicted data in the shared cache, in response to detecting the flag indicates the evicted data is not cacheable;

wherein said shared cache is shared by said graphics unit and a processor.

15. (Original) The method as recited in claim 14, further comprising resetting a count corresponding to each of said entries prior to rendering a new image.

16. (Original) The method as recited in claim 15, wherein said count is utilized in determining whether corresponding data is cacheable during the rendering of a second image.

17. (Original) The method as recited in claim 12, wherein said threshold is programmable.

18. (Original) The method as recited in claim 14, further comprising adjusting said threshold in response to a miss rate of processor data in the shared cache.

19. (Currently Amended) A graphics unit comprising:

a rendering unit;

a local cache configured to store graphics data; and

a prediction unit, wherein the prediction unit is configured to:

partition images to be rendered into a plurality of subset areas;

track the number of times data corresponding to each of the subset areas is considered during the rendering of a first image; and determine for each of the subset areas whether the corresponding subset area data is cacheable in a shared cache.

20. (Original) The graphics unit as recited in claim 19, wherein the prediction unit is configured to indicate a corresponding subset area is cacheable if the number of times data corresponding to the subset area was considered during the rendering of the first image exceeds a threshold.

21. (Original) The graphics unit as recited in claim 19, wherein the prediction unit is configured to indicate a corresponding subset area is cacheable if the number of times data corresponding to the subset area was considered during the rendering of the first image was high relative to other subset areas of the image.

22. (Original) The graphics unit as recited in claim 20, wherein the prediction unit includes storage which includes an entry for each of said subset areas, each entry including:

a count indicative of a number of times data corresponding to the subset area is considered during the rendering of an image.

23. (Original) The graphics unit as recited in claim 22, wherein the prediction unit is further configured to:

reset the count values for each of the subset area entries prior to rendering a new image; and

increment an entry count in response to performing a read of data corresponding to said subset area entry.

24. (Currently Amended) The graphics unit as recited in claim 19, wherein during the rendering of a second image, the graphics unit is further configured to store data evicted from the said local cache in a shared cache only if an indicator which corresponds to the data evicted from the local cache indicates the data evicted from the local cache is cacheable.

25. (Original) The graphics unit as recited in claim 20, wherein the threshold is programmable.

26. (Original) The graphics unit as recited in claim 24, wherein the threshold is adjusted in response to a miss rate of processor data in the shared cache.